

## REMARKS

### General Remarks

The applicant thanks the Office for its careful consideration of the previous response. The applicant likewise thanks the Office for its kind reminder to sign and date all correspondence. The applicant notes that the previous response was dated on page two and each subsequent page of the document, and was signed in accordance with 37 C.F.R. 1.4 (d)(2).

### Claims Rejections - 35 USC §102(b)

The Office rejected claims 1-3, 6-28 under 35 U.S.C. 102(b) as being anticipated by US Patent No. 5,752,032 issued to Keller et al. A rejection based on anticipation requires that a single reference teach every element of the claim (MPEP § 2131). "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Or stated in another way, a "claim is anticipated only if each and every element as set forth in the claim is found, . . . described in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

The applicant notes that in paragraphs 8 and 9, the Office rejects claims 4 and 5, which were both canceled in the previous Office Action Response.

The applicant respectfully suggests that the Office may find the following definitions commonly used in the relevant field of distributed computing to be helpful.

Physical device: "a specialized processing element (implemented as FPGAs, gate array, specialized integrated circuit, or parts of FPGAs". (*Para 0051, applicant's specification*)

Object: "individual units of running software that combine functionality and data, and that frequently (but not always) represent something in the real world."

<http://www.omg.org/gettingstarted/corbafaq.htm>

Software component: A **software component** is a system element offering a predefined service and able to communicate with other components. Clemens Szyperski and David Messerschmitt give the following five criteria for what a software component shall be to fulfill the definition:

- Multiple-use
- Non-context-specific
- Composable with other components
- Encapsulated i.e., non-investigable through its interfaces
- A unit of independent deployment and versioning

A simpler definition can be: A component is an object written to a specification. It does not matter what the specification is: COM, Java Beans, etc., as long as the object adheres to the specification. It is only by adhering to the specification that the object becomes a component and gains features like reusability and so forth.

Software components often take the form of objects or collections of objects (from object-oriented programming), in some binary or textual form, adhering to some interface description language (IDL) so that the component may exist autonomously from other components in a computer.

[http://en.wikipedia.org/wiki/Software\\_component](http://en.wikipedia.org/wiki/Software_component)

The applicant respectfully submits that the '032 reference fails to disclose a field programmable physical device that is loaded with a field programmable content or program. The '032 reference describes a video controller, made up of special function hardware. Each hardware component is configured with a specific function and circuitry that is defined by the installed circuit board.

The shell module 72 selects a particular board driver 74, from among any number of variants, that corresponds to the major aspects of the particular video display controller 19. In practical terms, the major aspect of a particular video display controller 19 is the specific architecture of the integrated circuit graphics accelerator chip or chip set used in

the implementation of the controller 19. Thus, a single board driver 74 may correspond to a well defined family of specific variants of the display controller 19. Other board drivers can be constructed to correspond to other families of controllers of different definition.

(Col. 9, ll. 23-34)

One skilled in the art can appreciate that this is in fact substantially the reverse of the claimed invention. While in the claimed invention, the functionality and capabilities of the processor are programmable and thus reconfigurable to be compatible with the demands of the system, the '032 reference selects software that is compatible with the specific video processor hardware.

In further contrast, the '032 reference, to introduce reconfigurability to the video controller, discloses the "dynamic configuration of the device driver 50 by the selective support of a set of hardware interface modules." (Col. 9, ll. 36-37). This, the applicant suggests, while superficially similar to the applicant's device interface, is in fact dissimilar. The '032 reference goes on to explain that the "set of interface modules dynamically loaded into the device driver 50 is determined by the board driver 74 in correspondence to the specific sub-elements present in the implementation of the video display controller 19." (Col. 9, ll. 47-51) The applicant respectfully submits that this discloses a system quite distinct from that of the claimed invention. Again, the '032 reference selects software based on the needs of the hardware, rather than programming programmable hardware to operate within a software framework.

While the hardware may be equipped with registers for the programming of commands such hardware is not programmable in the same sense as a field programmable gate array or other such devices which are described by the term "programmable physical device" as defined in the specification. Specifically, the '032 reference recites a system wherein the board driver 74, rather than the application dictates the interfaces and controls of the "device". Specifically, the '032 reference limits its programming of its hardware to change configuration variables, not to substitute or install downloaded programmable content or programs changing the functionality of the device itself as with an FPGA or other specialized processing element.

Indeed, the applicant notes that the '032 reference fails to disclose the separation of common control of the physical device from the control of the downloaded capability. The '032 reference, rather than allowing for the substitution of programs, instead attempts to change functionality by changing the hardware or peripherals attached to the system.

The applicant further submits that the '032 reference has failed to disclose a physical device object. While the '032 reference discloses physical devices, an object separate from a capability object or hardware module is not disclosed.

Further, the Office alleges with regard to Claim 2 that the '032 reference discloses use of physical devices that are chosen from the group of physical devices consisting of general purpose processors, specialized circuits and FPGAs, citing col. 6, ll. 48-63. In '032 Col. 6 lines 48-63 the '032 reference teaches that other peripherals may exist. In the cited section, the '032 reference teaches that the functions of the peripherals may be complex controllers. In the cited section, the '032 reference states that it supports controllers where the functional organization and operation of the controller is subdivided into smaller elements, that are directly programmable by the computer. However, the expression "programmable" in this context refers to setting values in a register interface, for each of the sub-elements of the controller, as described in a preceding paragraph at Col. 6 lines 19-22. The '032 reference discloses the ability to substitute various instances of the same type of controller (col 9 line15-26) (e.g. video controller), the context of the reference provides a clearer idea of its disclosure (col 9 lines 35-41) where the selection of hardware interface modules, corresponding "substantially to the individual sub-elements of a particular implementation of the video display controller" is intended. The '032 reference clearly does not envision that the function of the sub-elements would change, although absence of such a device could result in it not being loaded. *See* Col 9 51-60. Further clarifying the '032 reference's disclosure is the description of the initialization process, where in Col 14 line 46 – col15 line 1 it is clear the various sub-elements are fixed, being correlated to the board identifier created at time of manufacture of the controller, or must be determined by some analysis software. At no time in the initialization process so described Col. 13 line 5 – Col 19 line 11, does the download of any program to the physical device is

described—it simply was not part of the disclosure. In fact, the ‘032 reference teaches that the hardware interface modules are created after discovery of the board type of the physical hardware, which was put there when the controller was manufactured. The present invention in contrast, is specifically concerning the programmable nature of the physical device, and in fact describes in the specification, the download of a program for the programmable device.

The ‘032 reference further describes col 25 line 60-col 26 lines 6 that “the hardware interface objects are preferably functionally constrained to (s) support a well defined set of operations specific to a particular type of hardware sub-element, provide for at least the logical programming of relevant hardware interface registers to implement the function of each support operation, ... and finally(4) return certain data read directly or indirectly from the interface registers.” The ‘032 reference clarifies further, Col 26 line8 – 12, “In general, a call to a hardware interface object results in the generation of a reference to one or more registers of the interface registers, and a string of data values to be programmed into the referenced registers.”

The ‘032 reference is clearly not referring to the programming of an FPGA, as taught by Britton, et al. in 5,396,126 in the description of prior art col 1 lines 10-21, described as where the logic functions can be established by the user. This is a differentiating feature, and necessitates a novel approach, not envisioned by the ‘032 reference. In particular, the requirement to provide both objects for the capabilities within the programmable device (similar to the hardware interface objects taught by the ‘032 reference) and objects for the physical device itself.

The ‘032 reference continues the clarification of its disclosure (col 9 lines 35-41) as it describes the dynamic configuration of its device driver using loadable interface modules, where the selection of hardware interface modules, “correspond substantially to the individual sub-elements of a particular implementation of the video display controller.” The key here is that what is envisioned to change at run-time are the interfaces (hence the selection of various hardware interface [software] modules), not the overall function or purpose of the peripheral board or device.

The '032 reference clearly does not disclose or anticipate that the function of the sub-elements would change, although their absence could result in their not being loaded. Refer to Col 9 51-60 where the '032 reference teaches that the selection of the hardware interface [software] module of a particular function (e.g. clock) is a function of the [board] specific implementation of the corresponding hardware sub-element (Clock). What the '032 reference is teaching is all peripheral devices or boards of a certain function have the same general sub-elements, but differ by their specific implementation.

Further clarifying the '032 reference's intent is the description of the initialization process, where in Col 13 line 7 – 18 the '032 reference teaches of a board identifier located on the physical device, being placed typically during its manufacture (e.g. in ROM). The '032 reference continues to teach in Col 14 line 46 – col 15 line 1 the '032 reference teaches clearly that the various sub-elements are fixed, being correlated to the board identifier created at time of manufacture of the controller, or must be determined by some analysis software. Again this is taught by the '032 reference in Col 15 line 18 where it is the board identifier that is used to select the hardware interface [software] modules.

At no time in the initialization process so described (Col 13 line 5 – Col 19 line 11) does the download of any program to the physical device get described—because it was not envisioned. In fact, the '032 reference teaches the creation of the hardware interface modules are created after discovery of the board type of the physical hardware (Col 15 lines 18- 27), which was put there when the controller was manufactured.

The invention of the '032 reference provides an adaptive interface for this situation, but does not provide an adaptive interface when the function is created at run-time because the physical device is programmable, meaning non-specific and defined by the user.

The '032 reference describes (col 25 line 60-col 26 lines 6) that “the hardware interface objects are preferably functionally constrained to (1) support a well defined set of operations specific to a particular type of hardware sub-element, (2) provide for at least the logical programming of

relevant hardware interface registers to implement the function of each support operation, ... and finally(4) return certain data read directly or indirectly from the interface registers.” The ‘032 reference is teaching here that the physical device is providing a specific functionality.

As regards programmability, the ‘032 reference is clarifying here that it is disclosing setting values in hardware registers, not changing functionality. the ‘032 reference clarifies further, Col 26 line8 – 12, “In general, a call to a hardware interface object results in the generation of a reference to one or more registers of the interface registers, and a string of data values to be programmed into the referenced registers.”.

Further, the Office asserts regarding claim 10, the ‘032 reference teaches that said physical device interface object controls said physical device independently from a functionality performed by said physical device. In the cited reference (col 30 lines 34-45) the ‘032 reference describes only address indirection. That is the address of the hardware sub-element is computed based on the particular model of controller. In the present invention, the control of the physical device (as described in the specification para 0063) refers to those interfaces that are independent of the programmed load of the device. Since the ‘032 reference neither describes the ability for the controller to be a programmable device nor to have a programmed load, in the invention described by the ‘032 reference, there is no separate object nor need for one.

At least for those reasons stated above, the applicant respectfully submits that claims 1, 18, and 26, as amended, are patentably distinct from the ‘032 reference. As the remaining claims are dependant therefrom, the applicant respectfully submits that those claims are also patentably distinct. The applicant therefore respectfully requests that the Office withdraw its rejection of claims 1-3, 6-28.

Applicant believes the above amendments and remarks to be fully responsive to the Office Action, thereby placing this application in condition for allowance. No new matter is

Appl. No. 10/650,172  
Amdt. Dated: December 21, 2006  
Reply to Office Action of 8/21/2006

added. Applicant requests speedy reconsideration, and further requests that Examiner contact its attorney by telephone, facsimile, or email for quickest resolution, if there are any remaining issues.

Respectfully submitted,

**/Andrew P. Cernota, Reg. No. 52,711/**

Cus. No. 42716  
Maine & Asmus  
PO Box 3445  
Nashua, NH 03061-3445  
Tel. No. (603) 886-6100, Fax. No. (603) 886-4796  
Patents@maineandasmus.com

Scott J. Asmus, Reg. No. 42,269  
Andrew P. Cernota, Reg. No. 52,711  
Kristina M. Grasso, Reg. No. 39,205  
Attorneys/Agents for Applicant